

PENERAPAN METODE HEURISTIK (ALGORITMA IDA* DAN B&B) DALAM PEMECAHAN N-Queen Problem

Novi Penna¹, Sebastianus A.S. Mola², Meiton Boru³

^{1,2,3} Jurusan Ilmu Komputer, Fakultas Sains dan Teknik, Universitas Nusa Cendana

INTISARI

N-Queen problem merupakan suatu permasalahan dimana terdapat N buah bidak Ratu dalam permainan catur berukuran $n \times n$. Bidak ratu sebanyak N ditempatkan sedemikian rupa dalam papan catur dengan syarat Bidak Ratu tidak boleh saling memakan satu sama lain. Arah gerak makan pada *N-Queen Problem* sama dengan arah gerak bidak Ratu pada permainan catur secara umum yaitu bidak Ratu dapat bergerak ke arah horizontal kiri, horizontal kanan, vertikal atas, vertikal bawah dan juga arah diagonal, sehingga tidak boleh ada bidak Ratu yang saling berada dalam satu garis horizontal, vertikal, dan diagonal.

Metode Heuristik (*Heuristic searching*) merupakan salah satu metode yang dapat digunakan untuk menyelesaikan permainan *N-Queen problem* secara selektif, dengan memberikan solusi berupa jalur terpendek dan waktu yang efisien, sehingga memungkinkan pengguna dapat menyelesaikan permainan ini dengan baik, cepat dan relevan. Beberapa algoritma yang menggunakan fungsi heuristik adalah Algoritma *Iterative Deepening A** (IDA*) dan Algoritma *Branch & Bound* (B&B). Fungsi heuristik yang digunakan adalah dengan melihat banyaknya kotak yang masih kosong dan jumlah *queen* yang belum diletakkan pada papan.

Tujuan dari pembuatan proyek akhir ini untuk mengimplementasikan pemecahan *N-Queen problem* menggunakan metode heuristik (B&B dan IDA*). Dari aplikasi ini dapat dilihat bahwa Algoritma IDA* dan B&B mampu memberikan jalur dalam penyelesaian *N-Queen problem*. Setelah pengujian berulang yang dilakukan dengan menggunakan 19 dataset, dapat dilihat perbandingan hasil antara Algoritma IDA* dan Algoritma B&B dimana Algoritma IDA* dapat menyelesaikan *N-Queen problem* dengan node 61% dan waktu 41% lebih baik dari algoritma B&B.

Kata Kunci : *N-queen problem*, Metode Heuristik, Algoritma IDA*, Algoritma B&B.

ABSTRACT

N-Queen problem is a problem which a *N-Queen pawn* in is place chess with $n \times n$ size. *N-Queen pawn* is a put in such away in chess board with under condition that the queen pawns do not attack each other. The attacking movement of *N-Queen problem* is similar to the way of the queen pawn attacking in chess. Commonly the queen pawn moves horizontally to left and right, forward and backward vertically and also diagonally, so there are no queen pawns in a line of horizontal, vertical and diagonal.

Heuristical searching is one of the method which can be used to solve the game of *N-Queen problem* selectively, by giving solution of the shortest time channel efficiently in order to able the user to solve this game well, fast and relevantly. Some algorithms that use heuristic is *Iterative Deepening algorithm A** (IDA*) and *Branch and Bound* (B&B) algorithm. The used heuristic function is by seeing the numbers of boxes which are empty and the number of queen which is not be put in board yet.

The aim of making this final project to implement the solving of *N-Queen problem* using heuristic searching (B&B and IDA*). From this implementation could be seen that IDA* and B&B algorithm is able to give channel in solving *N-Queen problem*. After a repetition of test by using 19 sheet of data, it is shown the comparison of result between IDA* algorithm and B&B algorithm which IDA* algorithm result the shorter channel in solving *N-Queen problem* based on the node 61%, and time 41% which better than B&B algorithm.

Keywords: *N-Queen problem*, *Heuristical searching*, *Algorithm IDA**, *Algorithm B&B*

I. PENDAHULUAN

Catur merupakan sebuah permainan (*game*) yang populer dari kalangan anak-anak hingga dewasa yang dimainkan oleh 2 orang pemain. Dalam permainan catur diperlukan suatu perlengkapan permainan yaitu papan catur dan biji catur/bidak. Didalam pertandingannya, juga dibutuhkan strategi

khusus untuk memenangkan pertandingan tersebut. Salah satu *problem* yang ada yaitu masalah *n*-ratu atau *n-queen problem*, yaitu permasalahan mengenai bagaimana cara meletakkan bidak *queen* sebanyak *N* pada papan berukuran $n \times n$ sedemikian sehingga tidak ada 2 bidak *queen* yang dapat saling memakan hanya dengan 1 langkah (1 gerakan).

N-Queen problem merupakan suatu permasalahan dimana terdapat *N* buah bidak Ratu dalam permainan catur berukuran $n \times n$. Bidak ratu sebanyak *N* ditempatkan sedemikian rupa dalam papan catur dengan syarat Bidak Ratu tidak boleh saling memakan satu sama lain. Arah gerak makan pada *N-Queen Problem* sama dengan arah gerak bidak Ratu pada permainan catur secara umum yaitu bidak Ratu dapat bergerak ke arah horizontal kiri, horizontal kanan, vertikal atas, vertikal bawah dan juga arah diagonal, sehingga tidak boleh ada bidak Ratu yang saling berada dalam satu garis horizontal, vertikal, dan diagonal.

Metode Heuristik (*Heuristik searching*) merupakan salah satu metode yang dapat digunakan untuk menyelesaikan permainan *N-Queen problem* secara selektif, dengan memberikan solusi berupa jalur terpendek dan waktu yang efisien, sehingga memungkinkan pengguna dapat menyelesaikan permainan ini dengan baik, cepat dan relevan. Sehingga akan diterapkan 2 metode pencarian heuristik yaitu Algoritma *Branch and Bound (B&B)* dan Algoritma *Iterative Deepening A* (IDA*)*.

II. MATERI DAN METODE

2.1 Kecerdasan Buatan

Artificial Intelligence (AI) atau kecerdasan buatan merupakan cabang dari ilmu komputer yang mendayagunakan komputer sehingga dapat berperilaku cerdas seperti manusia. Pernyataan tersebut juga dapat dijadikan definisi dari AI. Definisi ini menunjukkan bahwa AI adalah bagian dari komputer sehingga harus didasarkan pada prinsip-prinsip aplikasi dari bidangnya.

2.2 Permainan (game) Catur

Catur adalah permainan yang dilangsungkan di atas papan yang terdiri dari 8 lajur dan 8 baris kotak/petak berwarna hitam dan putih (atau terang dan gelap) secara berselang-seling. Permainan dimulai dengan 16 buah pada masing-masing pihak, yang disusun berbaris secara khusus pada masing-masing sisi papan catur secara berhadap-hadapan. Satu buah *queen* hanya bisa menempati satu petak. Pada bagian terdepan masing-masing barisan terdapat 8 pion, diikuti di belakangnya dua benteng, dua kuda (dalam bahasa Inggris disebut *knight*-ksatria), dua gajah (dalam bahasa Inggris disebut *bishop*-uskup), satu menteri atau ratu atau *ster*, serta satu raja.

2.2.1 *N-Queen problem*

N-Queen adalah bagaimana cara meletakkan bidak *queen* catur sebanyak *N* buah pada papan catur atau pada papan ukuran $n \times n$ sedemikian rupa sehingga tidak ada satu bidakpun yang dapat memangsa bidak lainnya dengan 1 gerakan.

2.3 Teknik-teknik dasar dalam pencarian

Pencarian atau pelacakan merupakan salah satu teknik untuk menyelesaikan permasalahan *Artificial Intelligence (AI)*. Keberhasilan suatu sistem salah satunya ditentukan oleh kesuksesan dalam pencarian dan pencocokan [7].

2.4 Algoritma Pencarian

Terdapat banyak metode pencarian yang telah diusulkan. Berbagai metode yang ada dapat dibedakan ke dalam dua jenis, yaitu Pencarian Buta (*blind search*) dan Pencarian Terbimbing (*heuristic search*) [12].

2.4.1 Pencarian Buta (*blind search*)

Pencarian Buta adalah metode pencarian solusi yang tidak memiliki informasi awal untuk mengarahkan pencarian dalam mencapai *goal state* atau *current state* (keadaan sekarang).

2.4.2 Pencarian Terbimbing (*heuristic search*)

Pencarian Terbimbing adalah metode pencarian yang mempunyai informasi awal tentang *cost* atau biaya dalam mencapai *goal state* dari *current state*.

2.5 Algoritma *Branch and Bound*

Algoritma *Branch and Bound* merupakan salah satu algoritma yang menggunakan fungsi biaya dan pada setiap *node* diberikan nilai batas atau *bound* [13]. Rumus yang digunakan oleh algoritma B&B [8] yaitu:

$$C(i) = f(i) + g(i) \dots\dots\dots (1)$$

dengan:

$C(i)$ = nilai *cost*

$f(i)$ = jumlah *queen* yang belum diletakan pada papan

$g(i)$ = jumlah kotak yang masih kosong

2.6 Algoritma *Iterative Deepening A**

Merupakan salah satu metode pencarian yang merupakan pengembangan dari algoritma *Depth first search* yang menjadi batasan dari setiap penelusuran, berupa nilai yang paling minimum. Rumus yang digunakan oleh algoritma *IDA** [12] yaitu:

$$F(n) = g(n) + h(n) \dots\dots\dots (2)$$

dengan:

$F(n)$ = total nilai

$g(n)$ = jumlah kotak yang masih kosong

$h(n)$ = jumlah *queen* yang belum diletakkan pada papan

2.7 Metode Perancangan Sistem

Perancangan sistem merupakan penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi.

2.7.1 *Unified Modeling Language*

Unified modeling language (UML) merupakan sebuah bahasa pemodelan standar atau kumpulan teknik-teknik pemodelan yang dipakai untuk menspesifikasi, menampilkan, mengkonstruksi dan mendokumentasi hasil kerja dalam pengembangan perangkat lunak berorientasi obyek [5].

- ***Use Case Diagram***

Use case diagram adalah sebuah penggambaran dari sekelompok urutan kegiatan yang dikerjakan oleh sistem untuk menghasilkan keluaran yang dapat dilihat oleh pengguna (*actor/user*) [3].

- ***Activity Diagram***

Diagram aktivitas adalah salah satu diagram kondisi dalam UML yang digunakan untuk menggambarkan rangkaian aliran aktivitas/alur kerja (*workflow*) proses-proses *use case*.

2.7.2 *Bagan Alir (Flowchart)*

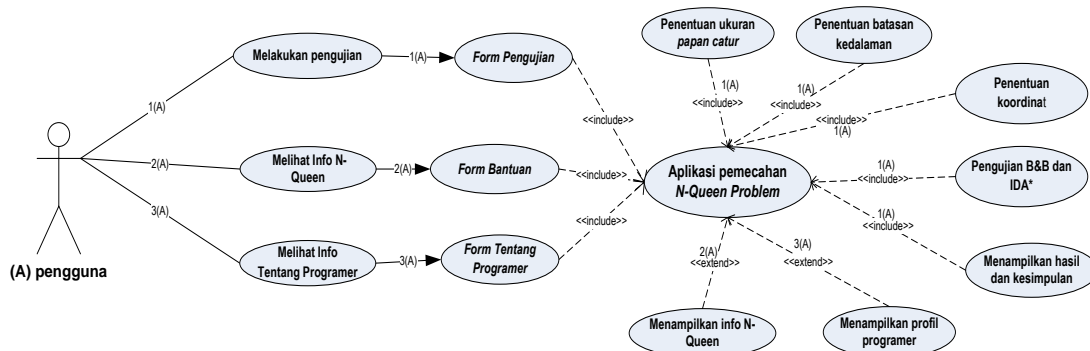
Flowchart Sistem merupakan bagan yang menunjukkan alur kerja atau apa yang sedang dikerjakan di dalam sistem secara keseluruhan dan menjelaskan urutan dari prosedur-prosedur yang ada di dalam system [4].

2.8 *Unified Modeling Language*

Unified modeling language (UML) merupakan sebuah bahasa pemodelan standar atau kumpulan teknik-teknik pemodelan yang dipakai untuk menspesifikasi.

2.8.1 *Use Case Diagram*

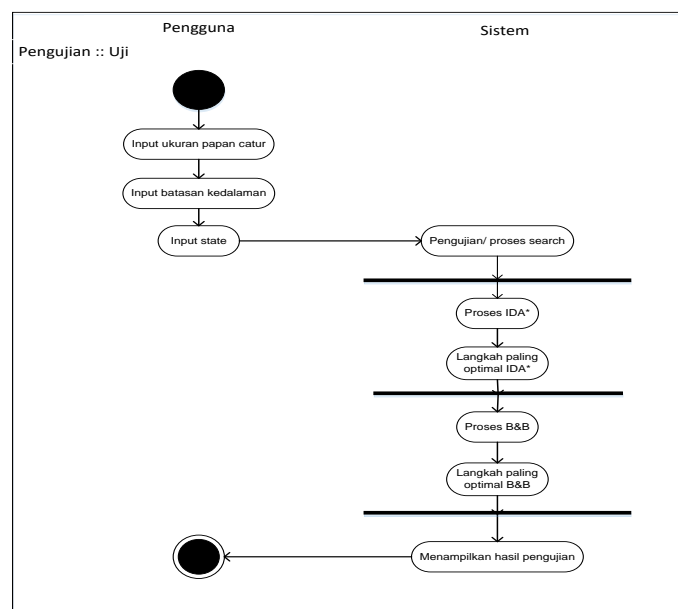
Use case diagram adalah sebuah penggambaran dari sekelompok urutan kegiatan yang dikerjakan oleh sistem untuk menghasilkan keluaran yang dapat dilihat oleh pengguna (*actor/user*) [3]. *Use case diagram* dapat dilihat pada gambar 1.



Gambar 1. Use case diagram sistem

2.8.2 Activity Diagram

Diagram aktivitas adalah salah satu diagram kondisi dalam UML yang digunakan untuk menggambarkan rangkaian aliran aktivitas/alur kerja (*workflow*) proses-proses *use case*. Activity diagram pengujian dapat dilihat pada gambar 2.

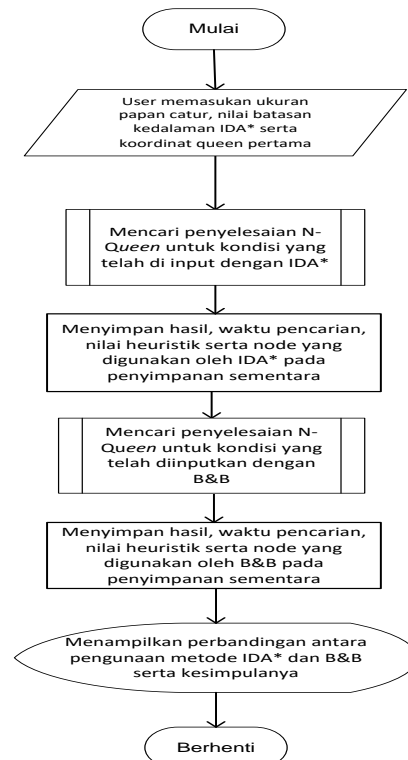


Gambar 2. Activity diagram pengujian

2.8.3 Flowchart program

Gambar 3 dapat dijelaskan seperti langkah-langkah dibawah ini:

- 1) Memasukan ukuran papan catur, batasan kedalaman IDA* serta koordinat *queen* pertama.
- 2) Menyelesaikan N-Queen problem untuk kondisi hasil dengan IDA* yang akan dijelaskan lebih lanjut pada gambar 3.8.
- 3) Sistem akan menghasilkan *output* berupa waktu pencarian, nilai heuristik serta *node* yang digunakan oleh IDA*.
- 4) Menyelesaikan N-Queen problem untuk kondisi hasil dengan B&B yang akan dijelaskan lebih lanjut pada gambar 3.9.
- 5) Sistem akan menghasilkan *output* berupa waktu pencarian, nilai heuristik serta *node* yang digunakan oleh B&B.
- 6) Menampilkan perbandingan serta *kenodean* antara penggunaan metode IDA* dan B&B.



Gambar 3. Flowchart program

2.9 Software yang Digunakan

Perangkat lunak (*software*) yang digunakan dalam pembuatan aplikasi ini adalah sistem operasi Windows 7, Program aplikasi Borland Delphi 2010.

III. HASIL DAN PEMBAHASAN

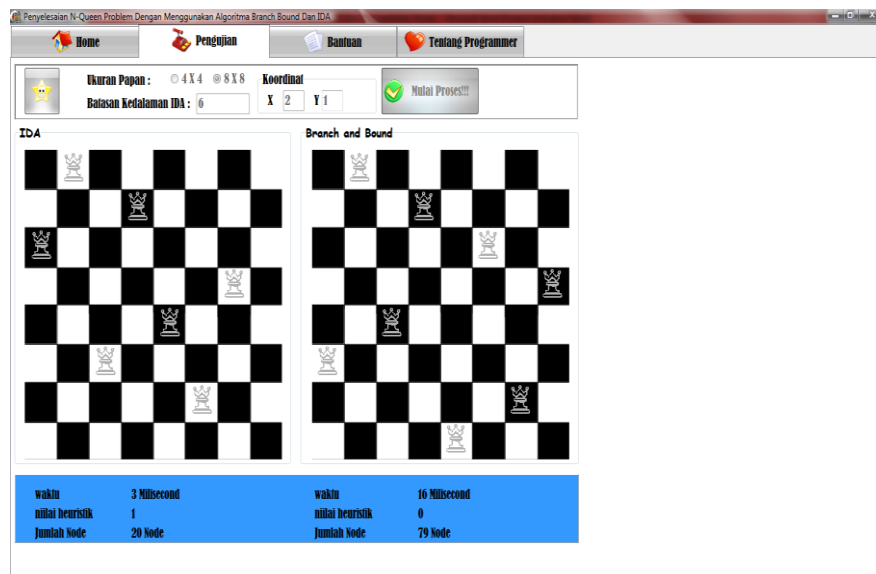
3.1 Hasil

Sesuai dengan tujuan dari penelitian ini maka dapat diketahui hasil unjuk kerja dari metode heuristik (Algoritma IDA* dan B&B) dalam aplikasi *N-Queen problem*, sehingga dapat dibuktikan bahwa kedua algoritma tersebut dapat digunakan dalam penyelesaian *N-Queen* dan dapat melihat perbandingan antara algoritma IDA* dan algoritma B&B [1].

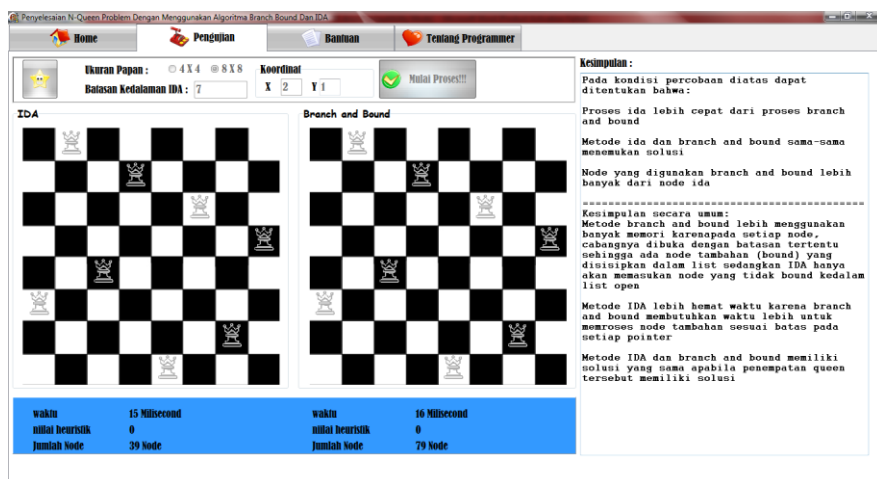
3.2 Pengujian Aplikasi

Pada *form* pengujian terlebih dahulu kita harus memilih ukuran papan catur yang akan diselesaikan, ukuran yang tersedia yaitu ukuran 4x4 dan 8x8. Kemudian masukan nilai batasan kedalaman yang berkisar dari angka 1 sampai 10 untuk algoritma IDA* yang akan digunakan untuk batasan dalam proses pencarian hasil (*goal state*). Setelah itu masukan koordinat x dan y dimana *queen* ingin ditempatkan [1]. *Form* pengujian dapat dilihat pada gambar 4.

Setelah kita mengisi semua dengan benar maka langkah selanjutnya adalah menekan tombol “Mulai Proses!”. Apabila kita lupa mengisi batasan kedalaman maka akan ada info pemberitahuan “silahkan mengisi batasan kedalaman terlebih dahulu”, hal tersebut juga terjadi jika lupa mengisi koordinat dan ukuran papan catur. Setelah menekan tombol “Mulai Proses!” maka hasil pengujian dari kedua algoritma dilihat dari waktu, nilai heuristik dan jumlah *node* akan ditampilkan. Kesimpulan dari kedua algoritma tersebut akan ditampilkan secara otomatis jika kedua algoritma telah menemukan solusi masing-masing. *Form* pengujian beserta kesimpulan dapat dilihat pada gambar 5.



Gambar 4. Tampilan form Pengujian



Gambar 5. Tampilan Form Pengujian beserta dengan kesimpulan

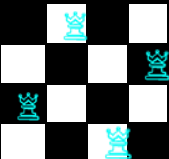
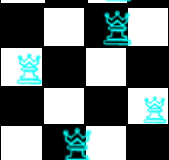
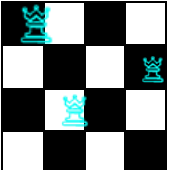
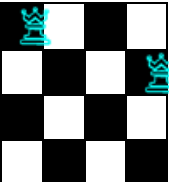
3.3 Pengujian dataset

Pengujian yang akan dilakukan terhadap aplikasi adalah dengan membandingkan kedua algoritma dengan parameter perbandingan yaitu jumlah *node* yang digunakan, nilai heuristik dan waktu pencarian. Untuk mempermudah melihat perbandingan dari hasil pengujian, maka dibuat tabel 1.

Pada tabel 1 dapat dilihat bahwa Algoritma IDA* lebih baik dari algoritma B&B baik itu dalam penggunaan node serta waktu yang digunakan untuk mencapai *goal*. Beberapa alasan yang dapat mendukung pernyataan tersebut adalah sebagai berikut:

- 1) Algoritma IDA* selalu menemukan solusi dengan waktu yang relatif singkat dari algoritma B&B. Dan apabila ada solusi lebih dari satu, maka IDA* akan menemukan solusi yang paling dekat dan membutuhkan waktu yang lebih sedikit.
- 2) Algoritma IDA* membutuhkan ruang memori yang lebih sedikit dari algoritma B&B.
- 3) Jumlah *node* yang digunakan dalam pencarian solusi minimal, berbeda dengan algoritma B&B.
- 4) Algoritma IDA* dipastikan dapat memberikan solusi yang diharapkan dan optimal.

Tabel 1. Perbandingan jumlah node IDA* dan B&B pada papan catur ukuran 4x4

| No | Data_Set | Koordinat | Algoritma IDA* | | | Algoritma B&B | | |
|-----------|---|-----------|----------------|--------------------|-----------------------|----------------|--------------------|-----------------------|
| | | | Jumlah Node | Nilai heuristik | Waktu (milisecond) | Jumlah Node | Nilai heuristik | Waktu (milisecond) |
| 1 |  | 2,1 | 4 | 0 | 0 | 10 | 0 | 0 |
| 2 |  | 3,1 | 4 | 0 | 0 | 10 | 0 | 0 |
| 3 |  <div>IDA*</div> | 1,1 | 4 | 1 | 0 | 11 | 4 | 0 |
| |  <div>B&B</div> | | | | | | | |
| Rata-rata | | | 4 | 0.333 | 0 | 10.333 | 1.333 | 0 |

IV. PENUTUP

4.1 Kesimpulan

Setelah mengetahui hasil perbandingan dari kedua metode heuristik (algoritma IDA* dan B&B), maka dapat ditarik kesimpulan sebagai berikut:

- Aplikasi ini dapat memberikan penyelesaian pada kasus *N-Queen problem* yaitu peletakan *queen* pada papan catur berukuran $n \times n$ tanpa ada *queen* yang berada pada lintasan yang telah dilalui oleh *queen* sebelumnya dan dapat dilihat cara kerja dari algoritma IDA* dan B&B dalam menyelesaikan *N-Queen problem*.
- Pada proses pengujian hanya diuji 19 dataset di karenakan 19 dataset ini telah mewakili semua dataset yang ada pada proses pengujian. Pada 19 dataset yang telah dilakukan dapat dilihat bahwa 18 *dataset* dapat memecahkan kasus dan memberikan hasil yang *complete* (selalu menemukan solusi jika solusinya ada), sedangkan hanya 1 *dataset* yang tidak menemukan solusi.
- Waktu yang digunakan pada algoritma IDA* lebih cepat 41% dari algoritma B&B sedangkan node yang digunakan pada B&B lebih banyak 61% dari algoritma IDA*, ini menunjukkan bahwa algoritma IDA* adalah algoritma terbaik dalam penggunaan waktu dan node yang digunakan lebih sedikit dibandingkan dengan Algoritma B&B.
- Untuk perhitungan pada ukuran 8x8, waktu pencarian yang dibutuhkan lebih lama dari ukuran 4x4. Ini disebabkan karena node yang dibutuhkan lebih banyak.

4.2 Saran

Beberapa saran yang dapat diberikan untuk membantu dalam pengembangan aplikasi ini lebih lanjut yaitu dapat dibuat *input* papan catur dengan ukuran yang lebih besar dan diharapkan dapat menggunakan metode-metode heuristik lainnya dalam penyelesaian *N-Queen problem* untuk mendapatkan hasil yang lebih optimal.

DAFTAR PUSTAKA

- [1] Penna, Novi. 2015. Pemecahan *N-Queen problem* menggunakan metode *heuristic search/informed search* (Algoritma *Branch and Bound* dan Algoritma *Iterative Deepening Search*). Kupang.
- [2] Anwar, Muhammad Nasrul. 2010. 8 *Puzzle dengan menggunakan algoritma Iterative Deepening Search (IDS)*, <http://journal.amikom.ac.id/index.php/TI/article/download/3808/1553>, diakses tanggal 12 Desember 2013.
- [3] Booch, Grady. 1999. *Visual Modeling With Rational Rose 2000 and UML*. Addison-Wesley. New Jersey.
- [4] Jogiyanto. 1999. Analisis dan Desain Sisem Informasi. Andi Offset. Yogyakarta.
- [5] Kadir, Abdul. 2003. *Pengenalan Sistem Informasi*. Andi Offset. Yogyakarta.
- [6] Kusumadewi, Sri. 2003. *Artificial Intelegent*. Graha Ilmu. Yogyakarta.
- [7] Kusumadewi, Sri. 2007. *Kecerdasan Buatan (Teknik dan Aplikasinya)*. Graha Ilmu. Yogyakarta.
- [8] Munir, Rinaldi. 2004. *Algoritma Branch and Bound*. Bahan Kuliah ke-11 IF2251 Strategi Algoritmik. <https://www.google.com/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF8#q=rinaldi+munir+algoritma+branch+and+bound+bahan+ku+liah+ke+11+IF2251+strategi+algoritmik>, diakses tanggal 02 Februari 2014.
- [9] Pertiwi, Nursyamsiah. dkk. 2006. *Penerapan Algoritma BFS, DFS, DLS dan IDS Dalam Pencarian Solusi Water Jug Problem*, <https://www.google.com/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=nursyamsiah+pertiwi+penerapan+algoritma+bfs,+dfs,,dfs+dan+ids+dalam+pencarian+solusi+water+jug+problem&spell=1>, diakses tanggal 02 Februari 2014.
- [10] Rumbaugh, James, dkk. 1999. *The Unified Modeling Language Reference Manual*. Addison-Wesley. New Jersey.
- [11] Setiawan, Christian Yulius. 2008. *Implementasi Metode Algoritma IDA* untuk Pencarian Jalur Terpendek Dengan Studi Kasus Yellow Pages Kota Bandung Menggunakan Teknologi WAP*. Bandung.
- [12] Suyanto. 2011. *Artificial Intelligence*. Bandung. Informatika.
- [13] Tandoe, Arie. 2012. *Penerapan Pohon dengan Algoritma Branch and Bound Dalam Menyelesaikan N-Queen Problem*, Makalah IF2091 Struktur Diskrit–SemI Tahun 2011/2012. <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2011-2012/Makalah2011/Makalah-IF2091-2011-074.pdf>, diakses tanggal 05 Februari 2014.